Engineering Applications of Artificial Intelligence 141 (2025) 109813

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Research paper

Triplet trustworthiness validation with knowledge graph reasoning

Geng Zhang, Yu-Jie Xiong *, Jian-Peng Hu *, Chun-Ming Xia

School of Electronic and Electrical Engineering, Shanghai University Of Engineering Science, No. 333, Longteng Road, Shanghai, 201620, China

ARTICLE INFO

Keywords:
Multiple-hop nodes reasoning
Triplets validation
Path graph inference
Graph reasoning

ABSTRACT

Knowledge graph is widely used in intelligent analysis and graph structure applications, which uses triplets to describe the relations and facts in the real world. In the process of incorporating external triplets into existing knowledge graph, it is vital to verify the trustworthiness level of triplet knowledge for building a comprehensive knowledge graph. In this paper, we establish a model for evaluating knowledge graph trustworthiness based on a triple strategy. The model quantifies the meaning of entities and relations expressed in the knowledge graph and obtains the quantification of triple trustworthiness measurement. And it integrates the internal semantic information of triplets and the global information of the knowledge graph, including entity-level, relation-level, and graph-level trustworthiness measurement, and finally uses multi-layer perceptron fusion to obtain the final score. This paper analyzes the effectiveness of the model output trustworthiness values and conducts error detection experiments in five real-world knowledge graph datasets. Experimental results show that compared with other models, our model has achieved significant effects.

1. Introduction

Knowledge Graphs (KGs), one of the fundamental trends driving the next generation of technologies, have now evolved into a new form of knowledge representation and the foundation of a variety of applications ranging from generic to specific industrial usage cases (Abu-Salih, 2021). They are widely used to describe the relationships between entities in the real world. Generally, entities serve as nodes and different relationships serve as edges, forming a network with a huge amount of knowledge (Hogan et al., 2021). Due to the uneven quality of data sources, the complex logic of semi-structured and unstructured data, effectiveness of the generated KG construction is affected, it will inevitably produce some noise errors and conflict triplets and has a influence on some tasks in the downstream of the KG (Hogan et al., 2021). Therefore, it is vital to verify the trustworthiness of external triplets before they have been added to the existing KGs. And it also improves the quality of constructing KGs and providing reliable and high-quality kGswith practical application value for its related tasks (Hogan et al., 2021).

KG is usually composed of triples (h, r, t), and its elements refer to head entity, relation and tail entity. Quantifying the authenticity of triplets is represented by the probability of trustworthiness. Generally speaking, the high trustworthiness of the triplet indicates that it is trusted and relevant to the KG. On the contrary, the low trustworthiness of the triplet indicates that it is not trusted, which contains head error, relation error and entity error. By detecting the quality of external triples, high-quality triplet knowledge can be discovered to ensure the quality of the KG, and it can be selected for KG path reasoning and question answering (Nguyen et al., 2020; Shi et al., 2022).

Recently, some methods construct trustworthiness calculations for triplets through inherent rules (Heindorf et al., 2016). It is assumed that the content of the existing triplets are completely correct, which often leads to significant potential errors, resulting in low fault tolerance and robustness of the model in prediction trustworthiness (Lin et al., 2015a). Especially when there is less information about entity relationships, the effect of trustworthiness testing is further exacerbated. Therefore, Chen et al. (2021) introduced a confidence prediction model for Semi-Supervised Learning, which effectively addresses scoring functions associated with various embedding methods. Additionally, they successfully generated negative samples based on confidence levels. But how to use appropriate methods to measure triples is a challenging task (Xie et al., 2018).

The application of graph models in KG is also quite extensive. The RGCN Schlichtkrull et al. (2018) introduces technique of parameters sharing and enforced sparsity constraints, and it is applied in modeling relation data, demonstrating excellent performance in link prediction and entity classification tasks. The GAT model Velickovic et al. (2017) performs attention computations on any nodes in the graph, calculating features by aggregating neighbor nodes feature. And the excellent properties of attention mechanism in sentence level relation and event extraction, long-distance dependency modeling, so

* Corresponding authors.
E-mail addresses: m320121326@sues.edu.cn (G. Zhang), xiong@sues.edu.cn (Y. Xiong), mr@sues.edu.cn (J. Hu), cmxia@sues.edu.cn (C. Xia).

https://doi.org/10.1016/j.engappai.2024.109813
Received 9 August 2023; Received in revised form 21 April 2024; Accepted 2 December 2024
Available online 11 December 2024

some researchers Ahmad et al. (2021), Yang et al. (2021) applied the idea in their tasks. Some feature-based ranking algorithms are also applied to KGs.

The PageRank (Brin, 1998) algorithm was originally used as a calculation method for the importance of Internet web pages, and was used to rank web pages in the Google search engine. The basic idea of the PageRank algorithm is to define a random walk model on the directed graph (Lao and Cohen, 2010). That is, a first-order Markov modular chain, which is used to describe the behavior of random walks randomly visiting each node along the directed graph. Under certain conditions, the probability of accessing each node in the limit case converges to a flat distribution, then the stable probability value of each node at this time is the PageRank value. Similarly, in the KG, each entity node also measures the confidence of the triplet through the importance factor. The PPNP model Klicpera et al. (2019) uses the personalized PageRank algorithm for semi-supervised learning to compute node features. Due to its high computational complexity, some researchers have proposed using the APPNP algorithm (Klicpera et al., 2019) to approximate the computation of PageRank and solve the features in it, achieving better results on multiple graph datasets.

This paper studies the quality validation of the triplets, and uses the semantic information contained in the existing KG to build a trustworthiness measurement model combining the multi hop information. The contributions of this paper are as follows:

- In this paper, we propose Two-step Approximate Attention Personalized Propagation of Neural Predictions (named as **TAAPPNP**) features extractor for validation of triplets, and it enhances entity information and improve the construction speed for KG.
- We propose a relation feature extraction method on Path Graph GRU Networks (named as **PGGN**), which directly connected the relation features between entities, and it enhances the effectiveness of relation feature extraction.
- The triple combination strategies is proposed as the Triplets Trustworthiness with Multi-hop Nodes Model (named as **TTMNM**), which contains the TAAPPNP, RGCN and PGGN, the fusion of the three parts enhances the effectiveness of validation for triplets.

The article unfolds as follows: It commences by elucidating the rationale and overview of the study on triplet verification, prominently pinpointing three novel contributions that serve to facilitate a deeper grasp of the domain for the reader. Next, it offers a sweeping examination of pertinent literature to furnish readers with a robust contextual foundation. In the third section, we delve into and expound upon the research motivation of this paper in a comprehensive and detailed manner. In section four, an in-depth exposition is given on the computational techniques and theoretical underpinnings of the triplet verification model, meticulously outlining its design principles and progressive execution steps. The fifth section carries out extensive experimentation and evaluation of the model using several openly accessible web-based datasets, systematically validating and scrutinizing its effectiveness from various angles. The sixth section sees the practical application of the model to real-world data drawn from industrial manufacturing contexts, yielding and reporting on the empirical validation results. Culminating the discourse, we summarize the core findings and embark upon a reflection on prospective avenues for further research.

2. Motivation

As illustrated in Fig. 1, KGs serve as an efficacious means for constructing knowledge bases and play a pivotal role in the process of domain-specific knowledge base development. They visualize data in a graph format imbued with inherent semantic relationships, thereby enhancing comprehension and facilitating the utilization of relevant knowledge resources. The construction process of a domain KG typically encompasses three main stages: Initially, during the data preprocessing phase, a combination of manual annotation and automated filtering techniques transforms external unstructured data into structured datasets that are more amenable to machine interpretation. Subsequently, this phase is further divided into three steps: Firstly, the post-preprocessed data undergoes knowledge categorization, culminating in large-scale datasets categorized into different classes. Secondly, entity relation extraction is performed on the classified data, furnishing high-quality triplet data for downstream tasks within the KG. Lastly, the extracted triplets undergo rigorous verification to ensure they meet the content requirements of the current KG under construction, ultimately generating a comprehensive and accurate triplet-based knowledge repository. This paper focuses on the critical issue of verifying new triplets within KGs, expanding the content of existing KGs by ensuring that freshly derived triplets are consistent with those already present. Diverging from entity linking tasks, our method targets validation of triplets produced by entity-relation extraction models, as well as externally introduced unknown triplets, aiming to ascertain their logical coherence with the pre-established KG and identify potential types of triplet errors. To achieve this objective, we first manually construct a small-scale KG as the preliminary knowledge input for the model, which is then used to generate feature vectors for evaluating the credibility of yet-to-be-introduced unknown triplets. By following this construction process, we eventually develop a more extensive domain-specific KG.

3. Related work

There are two common methods to build a KG (Ji et al., 2021; Weikum et al., 2021). The first one involves starting from scratch and constructing a new KG based on the accumulated data. This method has the advantage of being quick and easy to scale up. However, due to the assumption that the knowledge of triplets is correct, there is no existing knowledge is used as prior information, so it easily leads to the accumulation of errors in downstream applications. The second method involves updating and augmenting an existing KG with new data by inserting related triplet information. This approach can ensure the completeness and real-time relevance of the existing data while also enabling effective utilization of already-existing data. And some researchers come up with the bootstrapping method to build a KG (Xiao et al., 2016). However, ensuring the credibility and rationality of the added triplets poses a challenging research topic that warrants further exploration. Therefore, triplets trustworthiness validation has become a necessary part of building a KG (Cao et al., 2021). Trustworthiness is a measure of the credibility of the knowledge to be examined in the KG, which is a relatively comprehensive dimension that reflects the objectivity and verifiability of knowledge, especially for quality and error detection (Sidi et al., 2012). Generally speaking, the credibility of triplets based on trustworthiness testing is mainly divided into two categories: one is based on representation learning evaluation, and the other is based on combination strategy trustworthiness evaluation.

3.1. Representation learning evaluation

Trustworthiness assessment based on representation learning focuses on representation learning methods, generally improving the accuracy of trustworthiness assessment by mining potential rules and introducing feature information of triples. The translation model based on distance knowledge representation has become one of the important methods for KG validation. The TransE series model (Bordes et al., 2013; Wang et al., 2014; Lin et al., 2015b; Xiao et al., 2015; Ji et al., 2015) measures the rationality of the representation of triplets by calculating the distance between the head and tail entities through translation, nonlinear transformation, and other transformations. Some scholars have also proposed the CTransE model based on the clustering model (Li et al., 2020), which maps to the metric space according

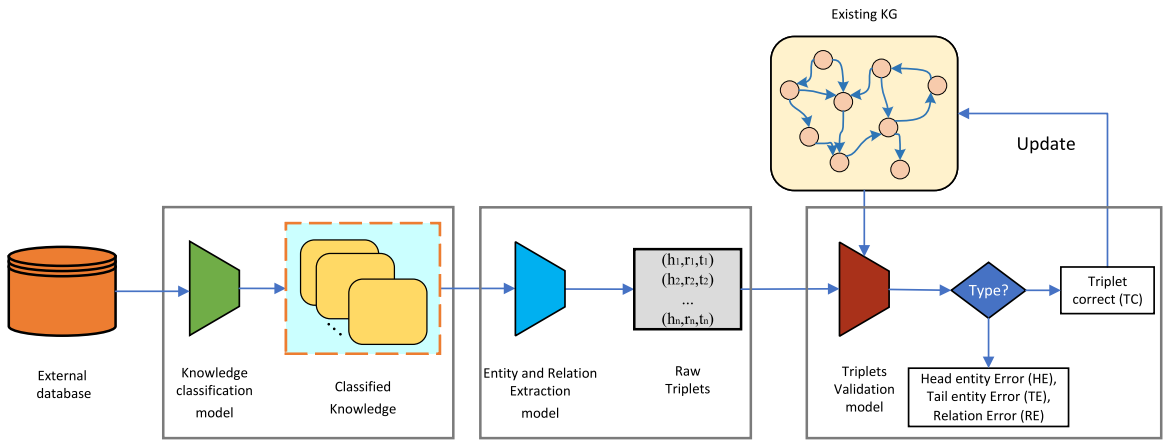


Fig. 1. The process of constructing domain KG.

to the rules and relationships between entities to measure the trustworthiness of the triple. According to the limitation representations of TransE, some researchers proposed TorusE (Ebisu and Ichise, 2018) model to change the mapping space from ordinary vector space to Lie group to solve the entity representation method. The PTransE model Lin et al. (2015a) proposes a path sorting method to construct a representation learning method in the KG, and applies it to noise detection tasks. Yet some researchers have proposed to divide the training of representation vectors into two steps: first, mapping entities into relational spaces, and then constructing translation relationships in the two projection spaces (Moon et al., 2017). Xie et al. (2020) proposed a new model named ReInceptionE, which combines the benefits of ConvE (Shang et al., 2019) and KBGAT (Nathani et al., 2019). It incorporates a relation-aware inception network that utilizes joint local–global structural information for KGE. The model first utilizes the Inception network to learn query embedding, aiming to further enhance the interactions between head and relation embeddings. It then suggests using a relation-aware attention mechanism to fortify the query embedding with local neighborhood and global entity information. Some researchers have keenly observed that, earlier models have often focused on enhancing the entity representation for multiple relationships, often overlooking the role of the single relationship vector. So they introduce TranS (Zhang et al., 2022), a novel transition-based KGE method. By replacing the traditional single relationship vector in the scoring pattern with a comprehensive relationship representation, TranS effectively addresses these issues.

3.2. Combination strategy evaluation

Based on combination strategy, trustworthiness is comprehensively evaluated from multiple perspectives, and combination of strategies can make more comprehensive use of trustworthiness to measure KGs. A literature Xie et al. (2018) proposes to combine local confidence, prior path confidence, and adaptive path confidence to obtain trustworthiness, which enables the model to distinguish noisy triplet data in the graph and achieve good results. But the researchers (Jia et al., 2019) have pointed out that there are two inevitable problems in the KG: (1) inevitably introducing noise and conflicts during construction; (2) In the upstream task of KG, the completeness of knowledge assumptions inevitably introduces potential errors. Due to these two issues, the some researcher Jia et al. (2019) comprehensively evaluates the KG from three aspects: entity-entity, relation-entity, and reachable path strategies. They propose KGTtm (Jia et al., 2019) to validate the knowledge in the graph, thereby improving the accuracy of representation learning and triplets validation. And Zhao and Liu (2019) proposed the SCEF model that has the similar structure of KGTtm, they comprehensively consider three parts: TransE, PRCA, and texture evidence, especially the impact of entity representation expansion on triplets validation.

Through the discussion of the above researches, although the model fully utilizes the internal structural features and constructs more features in the graph to verify the confidence of triplets, there are also problems with slow features construction and less semantic implications in entities and relations. This article constructs triple combined confidence evaluation strategy and proposes Triplets Trustworthiness with Multi-hop Nodes Model (named as **TTMNM**). We propose a comprehensive strategy for evaluating the confidence of triplets that leverages a pre-trained graph to incorporate the features of triplet semantics. Our model assesses the confidence of triplets by considering triple combination strategies, including entity-entity features within an internal multi-hop graph, entity-relation features within a relation graph, and reachable path features within a relation path graph. These different perspectives collectively contribute to a robust confidence validation method.

4. Our method

The task of the triplets validation needs to point out whether the triplet is credible and acceptable. We need to accurately obtain the trustworthiness value and error types of triples, so we transform it into a four classification problem rather than simply using trustworthiness to indicate whether the triplets is wrong. Given a triplet (h, r, t), where h, t represent the head entity and tail entity respectively, and r represents the relation, and the output of **TTMNM** is classified into four categories: triplet correct (**TC**), head entity error (**HE**), tail entity error (**TE**), and relation error (**RE**). The probability indicates the four categories trustworthiness of the triplet. **TTMNM** is a triple combination strategy graph model, which is divided into three parts: The first part is to measure the multi-hop features between entities (named as **TAAPPNP**), the second part is to measure the characteristics between entities and relations, the third part measures the features in relation path (named as **PGGN**). The following sections will explain the model structure and its characteristics. And the model structure is shown in Fig. 2. The symbols used in this chapter are shown in Table 1.

4.1. Input embedding

The input of the model is a batch of triplets. Then the triplets are transformed into triplet tokens and passed into the embedding layer, where they are indexed to obtain the triplet embedding vector. This step facilitates the calculation of features for the triples in graph recursion. As shown in the upper part in Fig. 2. So we can get the embedding vector of the triplet (h, r, t):

$$\begin{aligned} x_h &= E_h(h) \\ x_r &= E_r(r) \\ x_t &= E_t(t) \end{aligned} \tag{1}$$

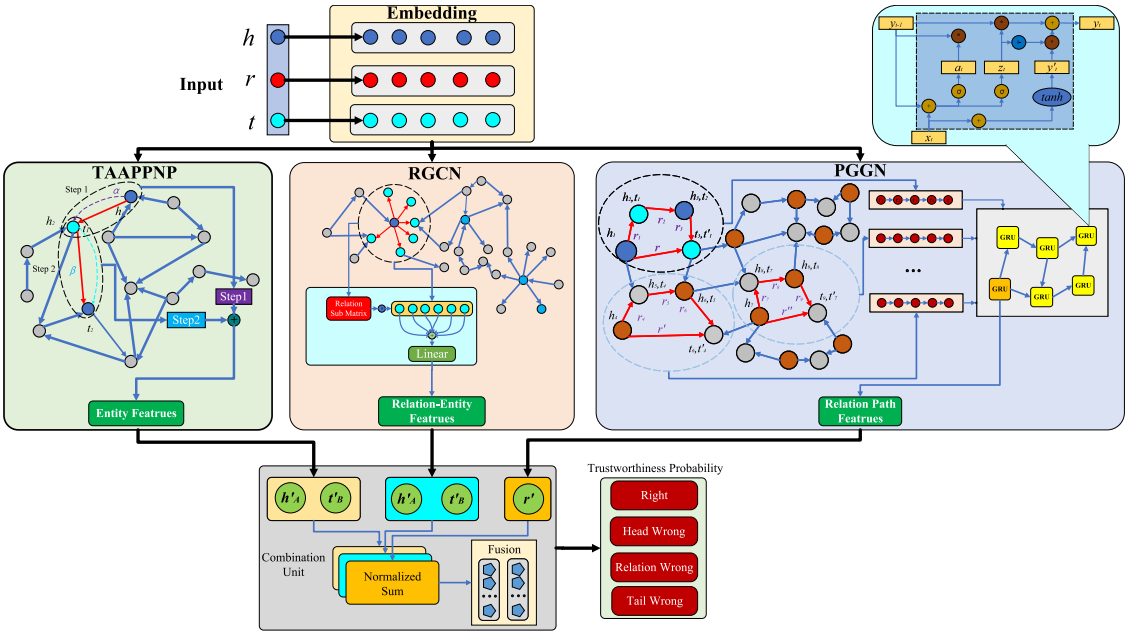


Fig. 2. The structure of TTMNM for triplets validation.

Table 1
Some symbols used in this chapter.

Symbols	Meanings
E_e	Entity embedding
E_r	Relation embedding
x_h	Head entity features vector
x_r	Relation features vector
x_t	Tail entity features vector
G	Graph relationships formed by (h, r, t) triplets.
$\pi_{ppr}(x_i)$	Feature vectors calculated through TAAPPNP graph
$\gamma_{i,j}$	TAAPPNP attention coefficient
$c_{i,r}$	Normalization parameter of RGCN

4.2. Two-step Approximate Attention Personalized Propagation of Neural Predictions

This paper considers the relationship between the head-tail and tail-head entities, and introduces a method of two hops in the entity graph node to measure the resource flow relationship between entity nodes. We propose the Two-step Approximate Attention Personalized Propagation of Neural Predictions (**TAAPPNP**) for entity feature extraction with two hops. The original APPNP (Klicpera et al., 2019) is defined as follows:

$$\pi_{\text{ppr}}\left(x_{i+1}\right) = (1-\alpha)\, A\pi_{\text{ppr}}\left(x_i\right) + \alpha x_i \tag{2}$$

We apply the idea of two hops. We take x_i, x_{i+1} as the initial head entities starting with two hops, x_{i+1}, x_{i+2} as the tail entities. And then we update the characteristics of resource flow between the two paths through approximate calculation. The calculation is expressed in the following form:

$$\begin{aligned}\pi_{\text{ppr}}\left(x_{i+1}\right) &= (1-\alpha)\, A_\alpha \pi_{\text{ppr}}\left(x_i\right) + \alpha x_i \\ \pi_{\text{ppr}}\left(x_{i+2}\right) &= (1-\beta)\, A_\beta \pi_{\text{ppr}}\left(x_{i+1}\right) + \beta x_{i+1}\end{aligned} \tag{3}$$

Where α, β respectively represents the first-hop and second-hop convergence coefficients, and x_i, x_{i+1}, x_{i+2} represents entities features respectively. We can understand the entire process of two hop node calculation from the first part in Fig. 2. In KG, first-hop refers to the head entity, and the second-hop refers to the tail entity. The algorithm calculates the resource flow of the head and tail entities with the entity's feature representation. The model particularly focuses on the two-hop relationship between entities to gauge the significance of resource feature values among them. Within KGs, a one-hop connection typically signifies a direct link from the head entity to a tail entity; however, extending to a two-hop relationship allows us to encompass a broader set of indirectly related tail entities. By considering such two-hop relationships, we can capture more intricate and global information flow patterns that reveal deeper structures of resource allocation or dependencies between entities.

Specifically, in the TAAPPNP framework, entity features undergo updates during the two-hop propagation process as demonstrated by the recursive formulas (as seen in Eq. (3)). This process integrates not only inherent attribute information of the entities themselves but also those of their immediate neighbors and even second-order neighboring entities. Thus, the two-hop propagation effectively simulates and quantifies the relative importance of resource flow between entities, refining the entity feature representations and enhancing the model's ability to reason about entity relationships. We can notice that some nodes are not connected, resulting in multiple subgraphs. Actually, the isolated nodes do not exist in TAAPPNP (as they are in the form of (head entity, tail entity)). The node $\pi_{ppr}(x_0)$ here is the first node entering the graph neural network, usually randomly selecting any node in the graph as the entry point for calculating graph features. However, when we calculate the relationship as the calculation node, there may be isolated points (mentioned in Section 4.4), and their features will not be updated. And this involves a two-hop stage, and we combine the model with an attention mechanism to solve this problem. Firstly, for node x, calculate the similarity coefficients between its neighbors j ($j \in N_i$) and itself one by one, and obtain e_{ij} through linear transformation layer:

$$e_{ij} = \sigma\left(\left[W_i \pi_{ppr}\left(x_{i+2}\right) \,||\, W_j \pi_{ppr}\left(x_{j+2}\right)\right]\right) \tag{4}$$

Secondly we calculate the attention coefficient:

$$\gamma_{ij} = \frac{\exp\left(LeakyReLU\left(e_{ij}\right)\right)}{\sum_{k\in N_i} \exp\left(LeakyReLU\left(e_{ik}\right)\right)} \tag{5}$$

Finally, the formula of the attention calculation is as follows:

$$\pi'_{\text{ppr}}\left(x_{i+2}\right) = \sigma\left(\sum_{j\in N_i} \gamma_{ij} W \pi_{\text{ppr}}\left(x_{i+2}\right)\right) \tag{6}$$

In particular, when $\beta = 0$, the model degenerates into APPNP. The two-hop relationship between entities describes the importance of resource feature values among entities.

4.3. Relational Data with Graph Convolutional Networks

To enhance the effect of entity validation, the paper uses Relational Data with Graph Convolutional Networks(**RGCN**) to enhance the effectiveness of data features. RGCN defines a multi-relational graph message propagation model, in which nodes v_i are updated as follows

$$x_i^{l+1} = \sigma\left(\sum_{r\in R}\sum_{j\in N_i^r}\frac{1}{c_{i,r}}W_r^{(l)}x_j^{(l)} + W_0^{(l)}x_i^{(l)}\right) \tag{7}$$

Where N_i^r represents the set of neighbor nodes whose relationship with node i is r, and $c_{i,r}$ is a regularization constant with $\left|N_i^r\right|$; the value of $W_r^{(l)}$ is an explicit conversion function, which converts the neighbor nodes of the same type using a parameter matrix $W_r^{(l)}$. From the middle part of Fig. 2, it can be seen that RGCN aggregates entity and relation features around nodes and outputs the final entity-relation features through a linear layer. In the RGCN, the inputs contain (head, relation, tail) triplets, and this heterogeneous network does not have isolated nodes.

The key advantage of RGCN is its ability to handle different types of edges within a graph by incorporating relational information. This is particularly important when dealing with real-world data, where entities are often interconnected in various ways and the relations can be highly complex. RGCN has been proposed as a useful framework for handling graph data that includes multiple types of relations, which enables a more accurate representation of the complex relationships between entities in real-world scenarios.

4.4. Path Graph GRU Networks

In order to obtain the information of entity path reasoning, the GGNN (Velickovic et al., 2017) is used here to obtain the relation paths information features. We called the layer as Path Graph GRU Networks (**PGGN**). This network uses GRU gating to calculate the feature information between relations in hidden layers:

$$\begin{cases} h_v^{(1)} &= \left[x_v^T, 0\right] \\ a_v^{(t)} &= A_{v:}^T\left[h_1^{(t-1)T}, \ldots, h_{|\mathcal{V}|}^{(t-1)T}\right]^T + b \\ z_v^{(t)} &= \sigma\left(W^z a_v^{(t)} + U^z h_v^{(t-1)}\right) \\ r_v^{(t)} &= \sigma\left(W^r a_v^{(t)} + U^r h_v^{(t-1)}\right) \\ \widehat{h_v^{(t)}} &= \tanh\left(W a_v^{(t)} + U\left(r_v^t \odot h_v^{(t-1)}\right)\right) \\ h_v^t &= \left(1 - z_v^t\right) \odot h_v^{(t-1)} + z_v^t \odot \widehat{h_v^{(t)}} \end{cases} \tag{8}$$

Node v first aggregates information from its adjacent nodes, where $A_{v:}^T$ is a sub-matrix of the adjacency matrix A of the graph, which represents the connection relationship of adjacent nodes. Similar to GRU's update function, it updates the node's hidden state using information from each node's neighbors as well as the previous time step. We essentially uses PGGN to generate an relations sequence to represent the characteristics of the relations (edges) path output. Since the output sequence is related to the feature relationship between each node, PGGN largely gathers the features of the relation path to enhance the effect in graph validation. Finally, the output of each node is expressed as the following form:

$$o_v = g\left(h_v^T, x_v\right) \tag{9}$$

From the part of PGGN in Fig. 2, we can see that a relation path between no more than four head-tail entities is retrieved from the graph. In the PGGN, as relationships involve isolated nodes, they are not subjected to update processing in such cases. These paths are sequentially arranged to form a relatio path graph, and the relationship features are calculated through PGGN. The graph relationship path model locks features from multiple relations nodes, increasing the effectiveness of feature output.

4.5. Model fusion layer

The fusion output layer is a core component of the network architecture proposed in this study, which integrates multi-dimensional information extraction and reasoning mechanisms, including multi-hop reasoning layer between entities, feature extraction layer based on entity and relationship features, and relationship path reasoning layer. During the training process, the network first utilizes three different graph neural network models to extract key information from the KG:

For multi-hop reasoning of entities, the TAAPPNP algorithm is used to process the graph G to capture complex multi-level associations between entities and obtain the multi-hop representation e_s of entities:

$$e_s = \text{TAAPPNP}(G) \tag{10}$$

The RGCN algorithm is used to compute on the graph GG to obtain the context feature representation e_c for each entity based on its adjacency relationships:

$$e_c = \text{RGCN}(G) \tag{11}$$

The PGGN algorithm is utilized to generate the relationship path reasoning feature r_g for entities by exploring and utilizing the relationship path information among entities:

$$r_g = \text{PGGN}(G) \tag{12}$$

After initializing these neural networks, for each triple (h, r, t) to be validated, the final validation result is computed through the following steps:

(1) First, combine the above three-dimensional feature vectors of the query entity into a fusion feature vector:

$$f = \left[h_s, t_s, h_c, t_c, r_g\right] \tag{13}$$

(2) Then, input this fusion feature vector into a multi-layer neural network FNN, which undergoes nonlinear transformation mapping to the output space to obtain the prediction:

$$p = \sigma(\text{FNN}(f)) \tag{14}$$

where σ is an activation function used to compress the target output values and ensure they fall within a valid range. FNN is a three-layer non-linear network, generally defined as follows:

$$\text{FNN}(x) = W_2 \cdot \text{ReLU}\left(W_1 x + b_1\right) + b_2 \tag{15}$$

In the above expression, W_1, W_2, b_1, b_2 are trainable parameters, and ReLU is the activation function. The role of the non-linear function is to integrate the three layers of entity-entity, entity-relation, and relation-relation relationships and map them into the classification space.

(3) Since this problem is defined as a four-classification task, where each sample needs to be classified into one of four categories, and the probability distribution of each category needs to conform to actual semantics, the SoftMax function is used to normalize the prediction values to obtain the final probability distribution:

$$s = \text{softMax}(p) \tag{16}$$

It is defined as a classification problem, and its final outputs are four categories, and their values represent the probability. The final loss function uses the cross entropy function to evaluate and update the gradient of the model:

$$L\left(s, s'\right) = -\sum_{i=1}^{S} s_i' \log(s_i) \tag{17}$$

Where s represents the probability distribution predicted by the model, and s' is the actual annotated label distribution, with S denoting the total number of classes. By minimizing this loss function, the model can more accurately learn the complex patterns of relationships between entities and improve the performance of KG verification based on this.

Table 2
The statistical of the datasets.

datasets	entities	relations	triples	train/test/valid
FB15K	14, 952	1, 346	1, 184, 426	966, 284/100, 000/118, 142
YAGO3–10	123, 183	38	2, 178, 080	2, 158, 080/10, 000/10, 000
WN18RR	40, 944	12	186, 006	173, 670/6, 068/6, 268
WN18	40, 944	19	302, 884	282, 884/10, 000/10, 000
FB15K-237	14, 542	238	620, 232	544, 230/35, 070/40, 932

5. Experiments

Through this section of experiments, we will verify the accuracy and consistency of the constructed KG triplets. The experimental results will help evaluate the quality and performance of the proposed model in this chapter, further validating the effectiveness and reliability of the proposed method. We conducted tests and analyses on different dimensions of the model, and provided detailed discussions on benchmark testing, ablation experiments, hyperparameter analysis, etc., to provide experimental evidence and support for the subsequent sections.

5.1. Datasets

This paper uses the different fields of public internet KG triplet datasets. The detailed information is as follows:

- **YAGO** (Suchanek et al., 2007): YAGO is an open source dataset that automatically extracts data from multiple sources such as Wikipedia, WordNet, and GeoNames. The paper uses the **YAGO3-10** dataset, and it is a subset of YAGO3 (which is an extension of YAGO), containing entities associated with at least ten different relations. **YAGO3-10** has a total of 123, 182 entities and 37 relations, 1, 179, 040 triples, most of which describe personal attributes such as citizenship, gender, and occupation.
- **FB15K** (Bordes et al., 2013): This dataset is a KG extracted from Freebase, which has abundant network resources, making FB15K the benchmark dataset as the KG research. FB15K-237 is a subset of the KG Freebase. In this article, FB15K and its subset FB15K-237 are used to evaluate the model;
- **WN18** (Bordes et al., 2013): This dataset captures approximately 41000 synonymous sets of 18 relationships from WordNet, resulting in 141442 triples. Research has found that a large number of test triples can be found in training sets with another or inverse relationship. Therefore, a new version of the dataset WN18RR has been proposed to address this issue.

All corresponding datasets are public available.[1] We conducted statistics on each dataset, and the statistical results are shown in Table 2. The distribution of different datasets can verify the superiority and inferiority of the model.

5.2. Experiment settings

The given datasets contain only positive examples and lacks negative examples, and the training process requires balancing the dataset to ensure the generalizability and stability of the dataset, as well as the accuracy and credibility of the test data, we have performed error examples generation on different datasets. Due to the presence of one-to-one, one-to-many and many-to-many scenarios within the triplets, take it into consideration, we generate errors with different distributions for each specific dataset. We firstly employ a general negative example generation method combined with positive example data to train the model. Consider an relation r_k that connects head entities with the number of N_{hr_k}, and it also connects tail entities with the number of N_{tr_k}. For datasets with significant differences in entity types, the generation process is as follows:

- **Entity Negative Examples**: Each triplet has its negative samples for its head entity and tail entity. For the same relation, when $N_{hr_k} > N_{tr_k}$, the relation and tail entity are fixed, and the head entity is randomly replaced. Conversely, when $N_{hr_k} \leq N_{tr_k}$, the relation and head entity are fixed, and the tail entity is randomly replaced. The newly added triples are not the original examples.
- **Relation Negative Examples**: For each triplet, there is an negative example of the relation. The head entity and tail entity are fixed, randomly replace the relation, and the added triples are not originally example data as well.

We setup several experiments that is trained on a server with NVIDIA RTX 3090 graphics card, 32 GB of RAM, a 120 GB external storage, and an i7-11700K intel processor, and we implement the network with PyTorch neural network framework.

This paper has conducted a large number of experiments and used grid search method to find the optimal model parameters. To ensure the effectiveness of comparing model training, the default parameter settings for entity and relation embeddings are set to 50 dimensions. The convergence iteration count for the graph TAAPPNP algorithm is set to 10, and its two-hops convergence parameters are set as $\alpha = 0.65$ and $\beta = 0.9$. The random weight parameters for dropout rate is set to 0.2. Since the TAAPPNP uses a multi-head graph attention module, the number of heads is set to 3 and the output layer dimension is set to 10. And the PGGN module is set to 10 layers.

The model presented in this paper is primarily designed to verify the consistency of triplets extracted after entity relation recognition with an existing KG. This task bears some resemblance to entity linking, yet its distinct focus lies in its objective: whereas entity linking aims to establish whether there exist triplet formations that adhere to factual reasoning, our study is dedicated to assessing the concordance of extracted triplets with a predefined KG. In contrast to entity linking tasks, which typically involve predicting subject entities $(?, r, t)$, relations $(h, ?, t)$, or tail entities $(h, r, ?)$, our research question centers on determining the accuracy of a given triplet $T = (h, r, t)$ based on the content within an established KG G. The output consists of four probability scores that reflect the confidence in their trustworthiness. As a comparative experiment, entity-relation representation methods are employed to validate the predicted triplets, with experiments incorporating four state-of-the-art entity-relation representation techniques included to demonstrate the thoroughness of the investigation.

5.3. Evaluation metrics

The evaluation of models represents an indispensable facts of the process, this is why we have elected to utilize the benchmark evaluation criteria prevalent in the realm of deep learning to rigorously measure the performance characteristics of the model we introduce herein. This paper adopts an ensemble of key performance indicators: the F1, Recall, Precision, and Accuracy, all of which contribute to providing a thorough assessment of the model's overall effectiveness. The details is as following:

- **Recall**: It measures the proportion of true positive cases correctly identified out of all actual positive cases, given by the following formula:

$$R = \frac{TP}{TP + FP} \tag{18}$$

- **Precision**: This metric represents the proportion of correctly predicted positive cases among those classified as positive, defined as:

$$P = \frac{TP}{TP + FN} \tag{19}$$

[1] https://github.com/louisccc/KGppler

- **F1**: This is the harmonic mean of Precision and Recall, which provides a single score that balances both the preciseness and completeness of the classification results. And calculated as:

$$F1 = \frac{2PR}{P+R} \tag{20}$$

- **Accuracy**: This is the overall correctness of the classifier, computed as the ratio of the total number of correct predictions (both true positives and true negatives) to the total number of instances, expressed as:

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \tag{21}$$

Here, TP represents True Positives,TN represents True Negatives, FP stands for False Positives, and FN denotes False Negatives. We have chosen these metrics because they are standard and widely accepted in the field of machine learning and information retrieval, providing comprehensive insight into different aspects of the classifier's effectiveness. We will expand upon this section to discuss why each metric is essential for evaluating our specific research problem and how they collectively contribute to the robustness of our findings.

5.4. Experiment results

Through this section's experiments, we aim to validate the accuracy and consistency of the KG triples constructed, with the experimental outcomes contributing significantly to the assessment of the quality and performance of the model proposed in this chapter. This further substantiates the effectiveness and reliability of the presented method. We have conducted thorough testing and analysis across various dimensions of the model, delving into detailed discussions on benchmarking comparation, ablation experiments, and hyperparameter analyses. These efforts provide solid experimental grounds and support for the discussions in subsequent chapters.

5.4.1. Benchmarks comparation

In this subsection, the benchmark testing of the model encompasses three key aspects for an overall comparison of its performance: a comparative analysis of the overall models, assessment of the convergence behavior of the model's loss, and a comparison of confusion matrices derived from the model's predictive outcomes. During the testing phase, the model is trained using optimal parameters to ensure the most rigorous performance evaluation.

Overall model comparison. The results as is shown in Tables 3–6. We compare several common models for triplets validation in KGs, and it can be observed that the triplet prediction scores are significantly improved due to the triple combination strategy. Such as Trans-Model series, InterHT (Wang et al., 2022), PairRE (Chao et al., 2021), TranS (Zhang et al., 2022), TripleRE (Yu et al., 2022) and so on. From the triple strategy, namely: TAAPPNP that reveals two-hops, RGCN that indicates entity-relation graph and PGGN that extract relation features. Therefore, the semantic characteristics between entities and relations are better identified. Compared with other models, the larger parameter size of our model makes the parameter information in triplets prediction more sufficient, and it greatly increases amounts of feature information for prediction, leads to an improvement of nearly 5%–15% in prediction accuracy, which is shown in Table 3. The Table 5 represents the precision rate of the model, while the Table 6 reflects the recall rate. It can be observed that in comparison to the baselines, the precision and recall rates are approximately 15%–20% higher, indicating that our model has excellent fitting performance. The F1 score is utilized as an evaluation metric for comprehensive assessment of the two values. From Table 4, we can see that the model surpasses the baselines by approximately 5%–20%, leading to a significant improvement.

Model convergence analysis. This paper meticulously examines the convergence behavior of the model throughout its training progression, as vividly depicted in Fig. 3. The training trajectory reveals an admirable degree of convergence for the model at hand. In contrast, alternative translation models with significantly smaller parameter counts manifest suboptimal convergence patterns. When juxtaposed against their counterparts operating under identical parameter configurations, the model presented herein demonstrates a superior convergence performance. This enhanced convergence efficacy can be attributed, in part, to the model's ability to transcend the limitations imposed by linear space transformations prevalent in comparative frameworks such as TransE and TransR. These latter architectures, due to their paucity of nonlinear operations, struggle to adequately accommodate complex, nonlinear feature computations. The model's consistent and robust convergence thus serves as a testament to its inherent reliability. Nevertheless, the graphical representation also discloses a tendency towards overfitting when the training regimen extends excessively. This phenomenon is particularly conspicuous in the case of the TranS model.

Confusion matrix analysis. Confusion matrix reflects the accuracy and robustness of the model from a visual perspective. We trained and predicted our proposed model on five datasets using the default optimal parameters. And we subsequently fine-tuned the negative triplets generation on this dataset, resulting in better results. The results are shown in Fig. 4. The "predict" column represents the model's predicted results, while the "real" column represents the labeled results of the dataset. The colors' intensity represents the corresponding accuracy of the prediction and labeling. From the figure, it can be seen that the prediction results perform well for the open datasets FB15K, FB15K-237, WN18, WN18RR, and YAGO3-10, achieved by generating different types of negative triplets to balance the training dataset. As shown in Fig. 4, the deep intensity color in the confusion matrix is concentrated along the diagonal, therefore it demonstrates the reliability of the model.

5.4.2. Ablation experiments

This paper conducts ablation experiments, as shown in Tables 7, 8, 9, 10. The ablation experiments compare the performance of the overall model with that of each strategy condition on triplets validation. From the overall perspective, the accuracy of the model shows better prediction performance under the triple strategy situation. Compared to the single TAAPPNP strategy, the performance of the combined strategy increased by 13.56%, 7.41%, 7.83%, 14.16%, and 7.49% on the five datasets, respectively; Compared with the RGCN strategy, the performance of the combination strategy has been improved by 12.96%, 1%, 29.7%, 9.25%, and 5.32%, respectively; However, compared to the PGGN strategy, for YAGO3-10, the F1 score of the combination strategy decreased by 2.22%, while other datasets increased by 0.37%, 16.6%, 4.28%, and 2.76%, respectively. For the YAGO3-10 dataset, the prediction performance is actually better under the relation-relation single strategy condition. This indicates that the PGGN provides better features under the triple strategy.

5.4.3. Hyperparameters analysis

TAAPPNP layer iteration effects. This article discusses the impact of the number of iterations in TAAPPNP layer inference. The number of iterations represents the convergence of features during the resource flow process. We conducted a convergence analysis of the corresponding number of iterations on five datasets, and the experimental results are shown in Fig. 5. The results show that for open-domain datasets FB15K and FB15K-237, the best performance is achieved when the number of iterations N reaches at 50–70, but when for the subset FB15K-237, the best performance is achieved at N = 30. For specific domain data, such as YAGO3-10 datasets, the best performance is achieved at N = 10, however, too many iterations may not achieve in good results. For the WN18RR and WN18 datasets, it is evident that the peak is reached at N = 50. The experimental results indicate that the number of iterations in the resource flow algorithm has a significant impact, and for different datasets, the number of iterations in the resource flow has an optimal peak, which leads to the best model prediction performance.

Table 3
The accuracy score prediction of model.

dataset	YAGO3-10	FB15K	FB15K-237	WN18	WN18RR
TransD (Ji et al., 2015)	0.8551	0.8311	0.7951	0.6375	0.4448
TransE (Li et al., 2020)	0.8699	0.8141	0.7802	0.6193	0.4395
TransH (Wang et al., 2014)	0.8776	0.8359	0.7858	0.5799	0.4357
TransR (Lin et al., 2015b)	0.8699	0.8155	0.7841	0.6361	0.4326
InterHT (Wang et al., 2022)	0.8739	0.8136	0.7726	0.6197	0.4423
PairRE (Chao et al., 2021)	0.8978	0.8941	0.8818	0.6487	0.4459
TranS (Zhang et al., 2022)	0.8801	0.8495	0.8074	0.6389	0.4271
TripleRE (Yu et al., 2022)	0.8957	0.8838	0.8603	0.7205	0.4511
TTMNM(Ours)	**0.9117**	**0.9107**	**0.8924**	**0.7775**	**0.5393**

Table 4
The F1 score prediction of model.

dataset	YAGO3-10	FB15K	FB15K-237	WN18	WN18RR
TransD (Ji et al., 2015)	0.8124	0.8219	0.7854	0.6221	0.3302
TransE (Li et al., 2020)	0.8112	0.8015	0.7711	0.5973	0.3366
TransH (Wang et al., 2014)	0.8339	0.8253	0.7675	0.5454	0.3261
TransR (Lin et al., 2015b)	0.8237	0.8029	0.7721	0.6164	0.3218
InterHT (Wang et al., 2022)	0.8305	0.8011	0.7644	0.5995	0.3353
PairRE (Chao et al., 2021)	0.8597	0.8837	0.8719	0.5949	0.3278
TranS (Zhang et al., 2022)	0.8225	0.8363	0.7873	0.6041	0.3236
TripleRE (Yu et al., 2022)	0.8541	0.8737	0.8475	0.7009	0.3573
TTMNM(Ours)	**0.8628**	**0.9012**	**0.8819**	**0.7712**	**0.4531**

Table 5
The precision score prediction of model.

dataset	YAGO3-10	FB15K	FB15K-237	WN18	WN18RR
TransD (Ji et al., 2015)	0.8201	0.8235	0.7894	0.6241	0.3479
TransE (Li et al., 2020)	0.7879	0.8022	0.7801	0.6052	0.3486
TransH (Wang et al., 2014)	0.8241	0.8308	0.7861	0.5571	0.3313
TransR (Lin et al., 2015b)	0.8066	0.8037	0.7802	0.6282	0.3645
InterHT (Wang et al., 2022)	0.8139	0.8038	0.7727	0.6064	0.3428
PairRE (Chao et al., 2021)	0.8414	0.8888	0.8735	0.6193	0.3374
TranS (Zhang et al., 2022)	0.8042	0.8435	0.7995	0.6226	0.3274
TripleRE (Yu et al., 2022)	0.8335	0.8787	0.8508	0.7039	0.3597
TTMNM(Ours)	**0.8415**	**0.9066**	**0.8819**	**0.7767**	**0.4564**

Table 6
The recall score prediction of model.

dataset	YAGO3-10	FB15K	FB15K-237	WN18	WN18RR
TransD (Ji et al., 2015)	0.8062	0.8204	0.7822	0.6213	0.4367
TransE (Li et al., 2020)	0.8511	0.8009	0.7666	0.5932	0.4578
TransH (Wang et al., 2014)	0.8511	0.8215	0.7552	0.5388	0.4175
TransR (Lin et al., 2015b)	0.8489	0.8021	0.7676	0.6099	0.4478
InterHT (Wang et al., 2022)	0.8573	0.7985	0.7615	0.5954	0.4628
PairRE (Chao et al., 2021)	0.8914	0.8807	0.8724	0.5869	0.4296
TranS (Zhang et al., 2022)	0.8595	0.8309	0.7784	0.5948	0.4436
TripleRE (Yu et al., 2022)	0.8971	0.8696	0.8448	0.6993	0.4672
TTMNM(Ours)	**0.9054**	**0.8969**	**0.8825**	**0.7678**	**0.5533**

Table 7
The f1 score of the model ablation.

	FB15K	YAGO3–10	WN18RR	WN18	FB15K-237
TAAPPNP:A	0.7681	0.7181	0.6746	0.6283	0.8048
RGCN:B	0.7741	0.7912	0.4559	0.6774	0.8265
PGGN:C	0.9000	0.8144	0.5869	0.7320	0.8570
TTMNM (A+B+C)	**0.9037**	0.7922	**0.7529**	**0.7699**	**0.8797**

Table 8
The accuracy score of the model ablation.

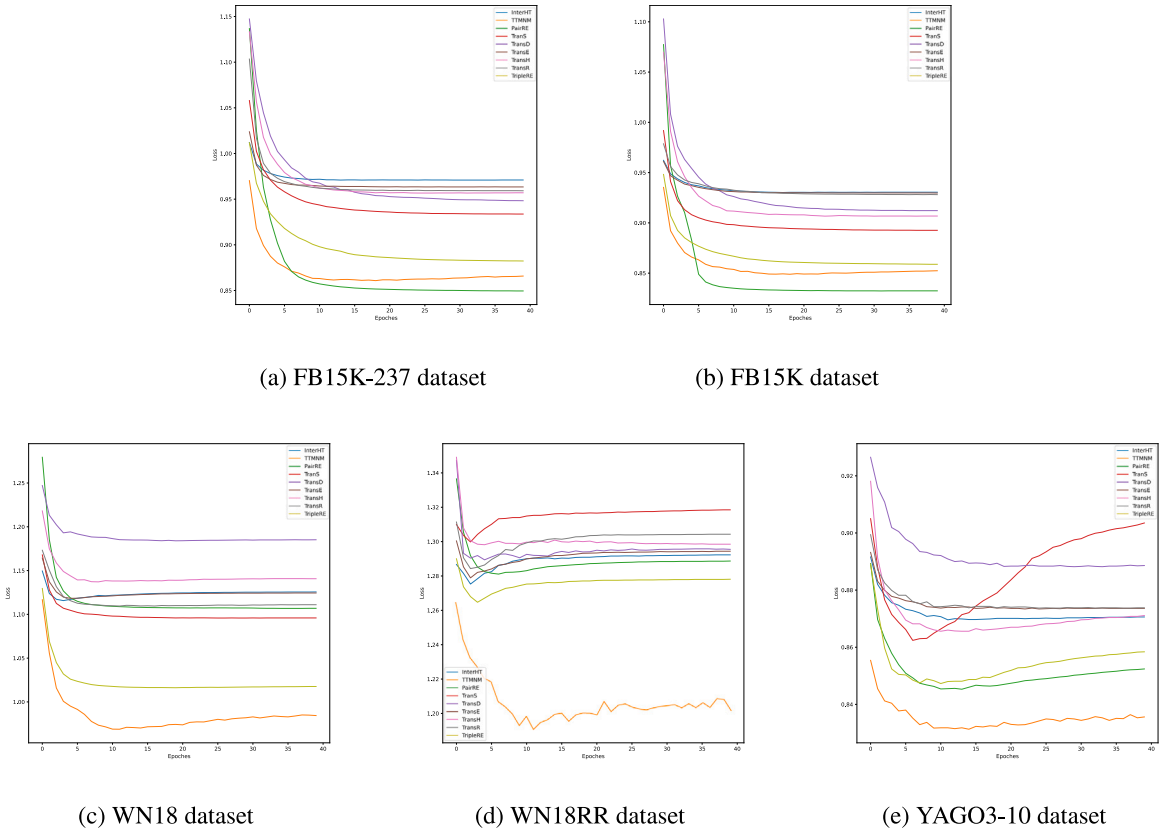
	FB15K	YAGO3–10	WN18RR	WN18	FB15K-237
TAAPPNP:A	0.7787	0.8865	0.6836	0.6403	0.8155
RGCN:B	0.7837	0.8624	0.5411	0.6843	0.8371
PGGN:C	0.9100	0.8677	0.6376	0.7342	0.8611
TTMNM (A+B+C)	**0.9124**	**0.9097**	**0.7599**	**0.7770**	**0.8887**

Table 9
The precision score of the model ablation.

	FB15K	YAGO3–10	WN18RR	WN18	FB15K-237
TAAPPNP:A	0.7613	0.9164	0.6849	0.6407	0.8100
RGCN:B	0.7672	0.8879	0.5424	0.6774	0.8310
PGGN:C	0.9024	0.8306	0.6175	0.7462	0.8587
TTMNM (A+B+C)	**0.9079**	**0.9341**	**0.7564**	**0.7798**	**0.8824**

Table 10
The recall score of the model ablation.

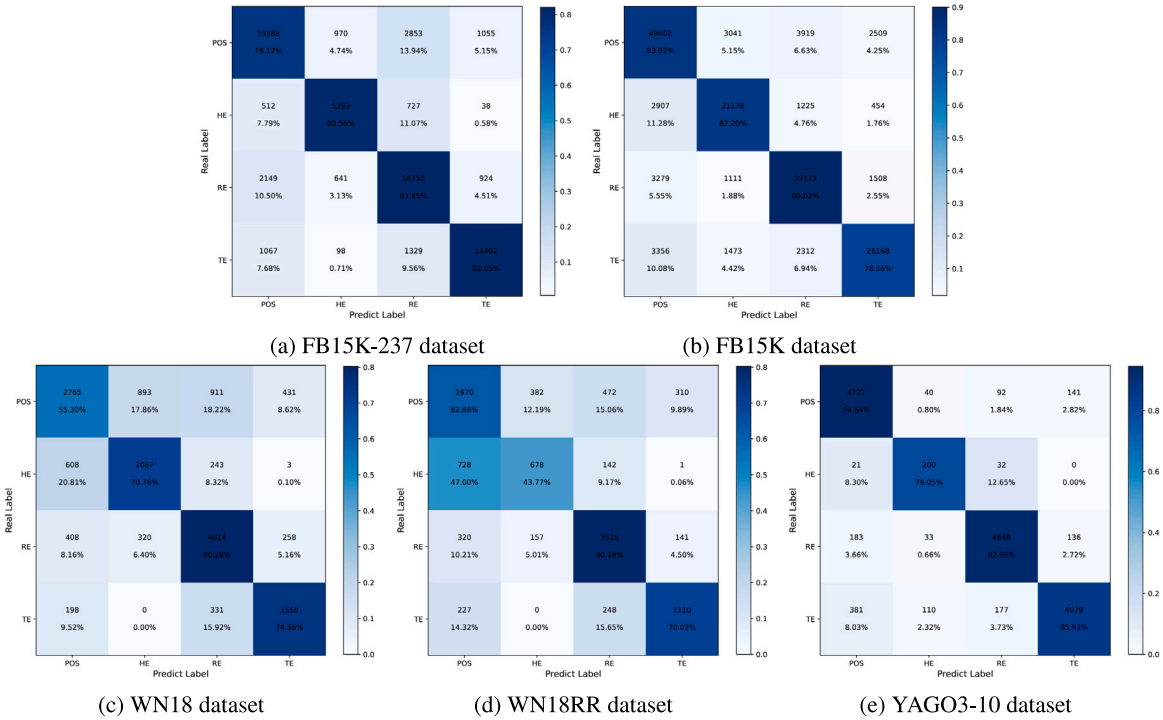
	FB15K	YAGO3–10	WN18RR	WN18	FB15K-237
TAAPPNP:A	0.7810	0.7010	0.6698	0.623	0.8014
RGCN:B	0.7873	0.7843	0.4604	0.6846	0.826
PGGN:C	0.8993	**0.8022**	0.5927	0.7297	0.8559
TTMNM (A+B+C)	**0.9011**	0.7571	**0.7497**	**0.7649**	**0.8777**



(a) FB15K-237 dataset (b) FB15K dataset

(c) WN18 dataset (d) WN18RR dataset (e) YAGO3-10 dataset

Fig. 3. TTMNM loss during training.

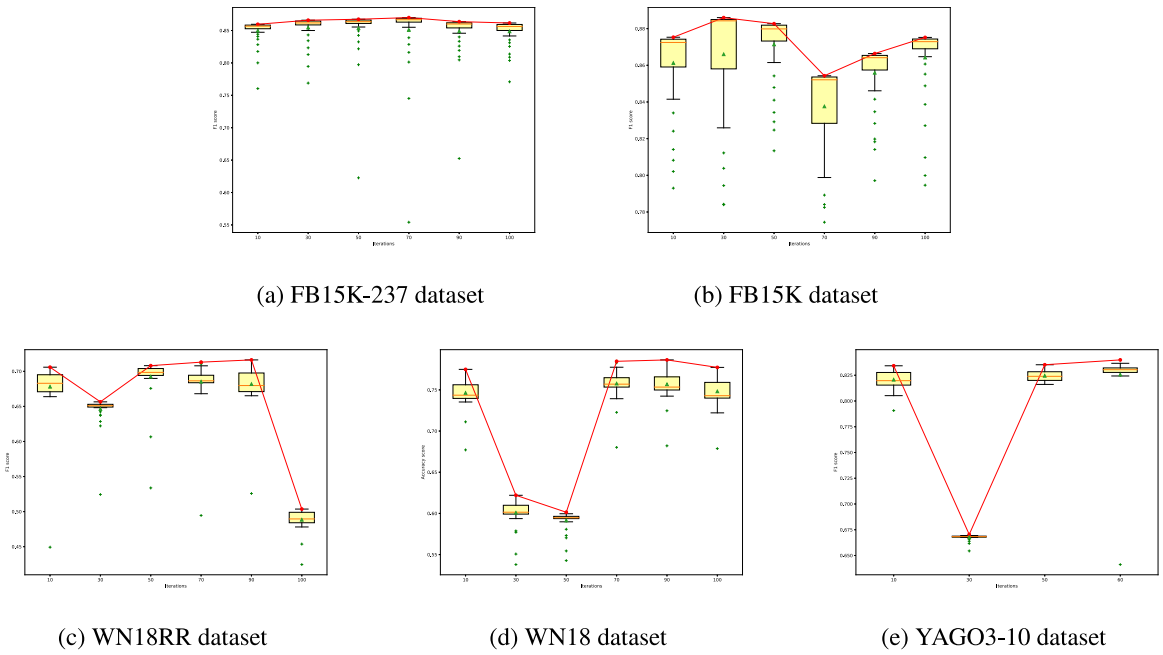


(a) FB15K-237 dataset (b) FB15K dataset

(c) WN18 dataset (d) WN18RR dataset (e) YAGO3-10 dataset

Fig. 4. TTMNM confusion matrix.

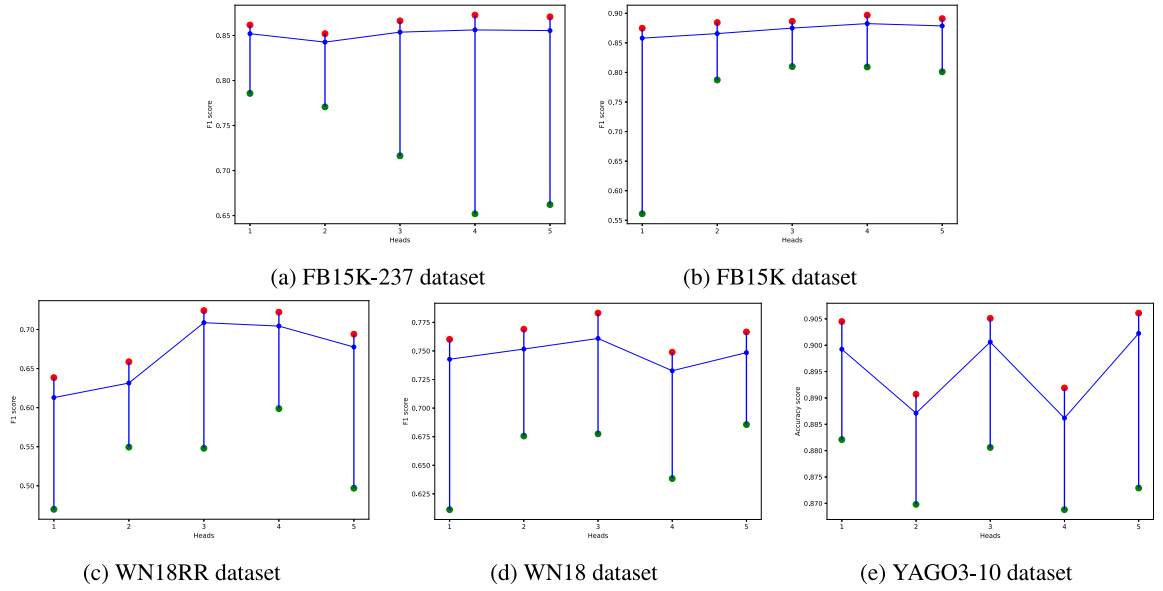
The effect of attention heads. The number of heads affects the feature extraction capability of the attention matrix, which in turn affects the F1 score of the final results. Therefore, this section focuses on studying the impact of different numbers of heads on the model's prediction ability. We conducted experiments on five datasets and set $N_h = 1, 2, 3, 4, 5$ to investigate the influence of the number of attention heads on the model's performance, the variation of F1 score is shown in Fig. 6. From the figure, it can be observed that each dataset corresponds to



(a) FB15K-237 dataset (b) FB15K dataset

(c) WN18RR dataset (d) WN18 dataset (e) YAGO3-10 dataset

Fig. 5. TAAPPNP iteration analysis.



(a) FB15K-237 dataset (b) FB15K dataset

(c) WN18RR dataset (d) WN18 dataset (e) YAGO3-10 dataset

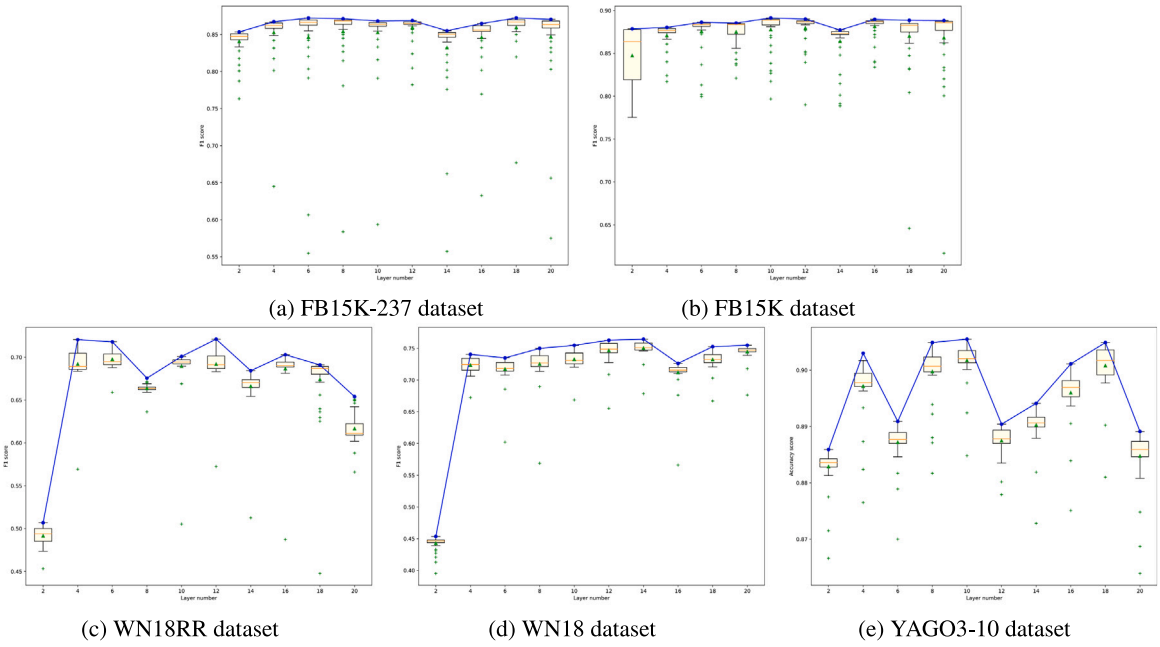
Fig. 6. Heads number of TAAPPNP analysis.

a peak value of F1 score. Analyzing the figure, it can be determined that the optimal number of attention heads for the five datasets is $N_h = 4, 4, 2, 3, 3$, respectively. This is due to the different characteristics of distribution exhibited by each dataset resulting in different peak results. The experiments demonstrate that the number of attention heads significantly affects the prediction results, but there always exists a corresponding peak value.

The effect of PGGN layers. The number of layers in the PGGN determines the level of relation feature extraction. Here, we demonstrate this point through several experiments. We conducted comparative analysis experiments on five datasets, where we set the number of layers in the PGGN as $N_l = 2, 4, 6, 8, 10, 12, 14, 16, 18, 20$. We plotted box plots to compare the model's predictive performance, and the variation of F1 score is shown in Fig. 7. From the data in the figure, it can be seen that the model's predictions are changed as the number of layers change, but there is always an optimal peak with best performance. For FB15K and FB15K-237, there is a smooth change in F1 score as the number of layers increases. For WN18RR, WN18, and YAGO3-10, there are more fluctuating changes in F1 score, but each dataset has a corresponding peak at a different position. It can be observed that the optimal number of layers for the five datasets are $N_l = 12, 12, 8, 12, 16$, respectively. The characteristics of the datasets distribution determine the differences in peak values. The experiment proved that the number of PGGN has an impact on the predicted score of the model to some extent.

5.5. Discussion

The distribution of the dataset has a significant impact on model training and prediction results. The generation of negative examples has a significant effect on the test sets of FB15K, FB15K-237, and



(a) FB15K-237 dataset (b) FB15K dataset

(c) WN18RR dataset (d) WN18 dataset (e) YAGO3-10 dataset

Fig. 7. PGGN layers analysis.

YAGO3-10 datasets in open domain scenarios. Because these datasets have a large number of relationships and significant differences between entity relationships, this method of generating negative examples is suitable for such datasets. For the WNRR datasets, the performance is moderate and also shows good results. However, for the WNRR18 dataset, there is a significant variation in performance. The relation prediction is relatively accurate, but there are errors in predicting the head and tail entities, which shown in Fig. 4(d). Based on the predicted results and labels of the data, as well as the semantic information, it can be seen that the inaccurate prediction of the head and tail entities is due to the existence of semantically similar phenomena in the WNRR18 dataset. In response to this issue, we have improved the negative example generation algorithm in our industrial datasets application. Specifically, we have modified the random entity replacement method used in the previous negative example algorithm to replace head and tail entities of different semantic types. This modification has led to the creation of prominent negative example data. By comparing statistics, it was found that there has been a significant improvement in the test results of generating negative examples using the above dataset. This negative case generation algorithm performs well when applied to industrial datasets.

6. Application

In this paper, we also apply our method to KG triplets validation in the industrial manufacturing domain. Due to the issues of missing and duplicated data sets in the industrial field, we extracted corpus from the text data of industrial manufacturing to provide data for KG triplets trustworthiness validation. We used the CasRel (Wei et al., 2020) model to perform entity-relation joint extraction on the collected industrial dataset, and obtained KG triplets dataset. Since there were errors in the extracted dataset, we manually checked and filtered the data, finally we get the KG corpus in a total of 10, 869 triples in the KG dataset, containing 6, 200 entities and 4 relations. Based on its inherent characteristics, we classified it into three datasets: train set, validation set, and test set, with a ratio of 6:1:1. Following the method mentioned in the previous section, we changed the way of replacing entities. The data distribution is shown in Fig. 8.

The model comparison is as shown in Table 11. Through the comparison of models, it can be seen that the model mentioned in this paper has the best performance, especially in terms of recall, precision, and F1 score. Compared to the baselines, the overall performance is improved by approximately 10%–20% The results demonstrates that the model has a good predictive effect in the field of industrial KGs triplets validation.

Table 11
The model comparison.

Model	F1	Accuracy	Precision	Recall
TransD (Ji et al., 2015)	0.4755	0.6251	0.4974	0.4744
TransE (Li et al., 2020)	0.5542	0.7381	0.5482	0.5621
TransH (Wang et al., 2014)	0.4588	0.6005	0.4682	0.4564
TransR (Lin et al., 2015b)	0.5233	0.6941	0.7607	0.5281
InterHT (Wang et al., 2022)	0.5564	0.7395	0.5515	0.5633
PairRE (Chao et al., 2021)	0.4897	0.6463	0.5495	0.4922
TranS (Zhang et al., 2022)	0.5141	0.6821	0.5096	0.5193
TripleRE (Yu et al., 2022)	0.5541	0.7373	**0.7914**	0.5616
TTMNM (Ours)	**0.7110**	**0.7455**	0.7506	**0.7294**

Confusion matrix further verifies and demonstrates the accuracy and robustness of the predicted results. It has good adaptability to KG data in the industrial field, as shown with the visualization in Fig. 9. It can be seen that the prediction is more accurate for relation errors and tail entity predictions, but weaker for head entities. The main reason is that the proportion of negative triples for head entities is not appropriate, which leads to a weaker prediction. The results of our proposed method are excellent for industrial datasets with incomplete data and weak entity with relation.

In the case studies detailed within Table 12, we thoroughly investigate the model's performance across numerous instances within an industrial dataset. The table presents eight illustrative examples, wherein 'Head' refers to the source entity, 'Relation' signifies the inter-entity relationship, and 'Tail' denotes the target entity. The 'Real Tag' column specifies the actual error categories, including true classifications (Triple Correct, TC), head entity errors (Head Entity Error, HE), relationship errors (Relationship Error, RE), and tail entity errors (Tail Entity Error, TE). The 'Pred Tag' column reflects the error type predicted by the model, each entry further accompanied by a color-coded score indicative of the model's prediction confidence, with deeper hues denoting lower confidence. The table shows that the model largely provides accurate predictions for triples like (Gear, part_of,

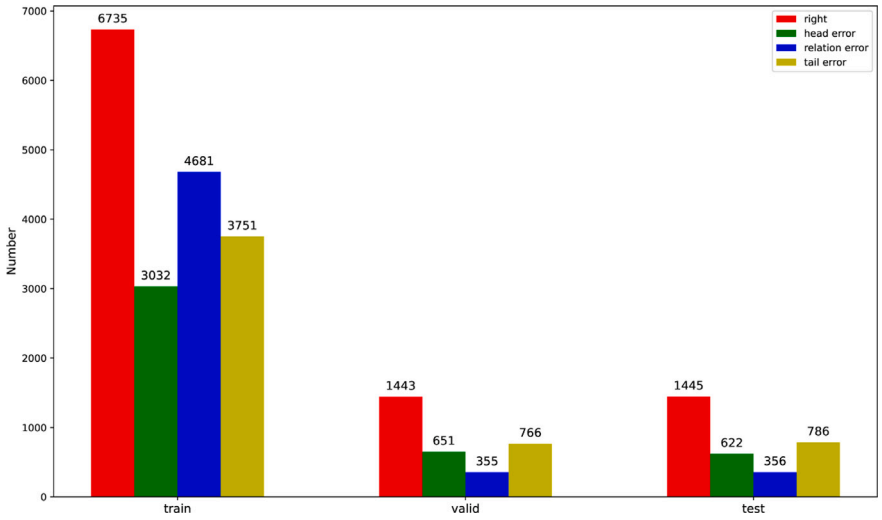


Fig. 8. The dataset distribution.

Table 12
The cases of the industrial dataset.

Heads	Relation	Tail	Real Tag	Pred Tag	TC	HE	RE	TE
Gear	part_of	Servo motor assembly	TE	TE (✓)	0.02	0.05	0.12	0.81
Iron core grounding wire	part_of	Bolt	TE	TC (✗)	0.56	0.02	0.01	0.41
Left A-pillar	part_failure	Smoke	TC	TC (✓)	0.41	0.12	0.12	0.35
Button "AUTO"	part_of	Stuck	RE	RE (✓)	0.09	0.17	0.63	0.11
Right-front wheel	part_failure	Noise	TC	TC (✓)	0.81	0.05	0.12	0.02
Sunroof	part_failure	Not working properly	TC	HE (✗)	0.27	0.66	0.03	0.04
First time airbag light	attr_failure	Alarm	RE	RE (✓)	0.01	0.03	0.91	0.05
Throttle valve	part_failure	Striped deformation	TE	TC (✗)	0.55	0.03	0.03	0.39

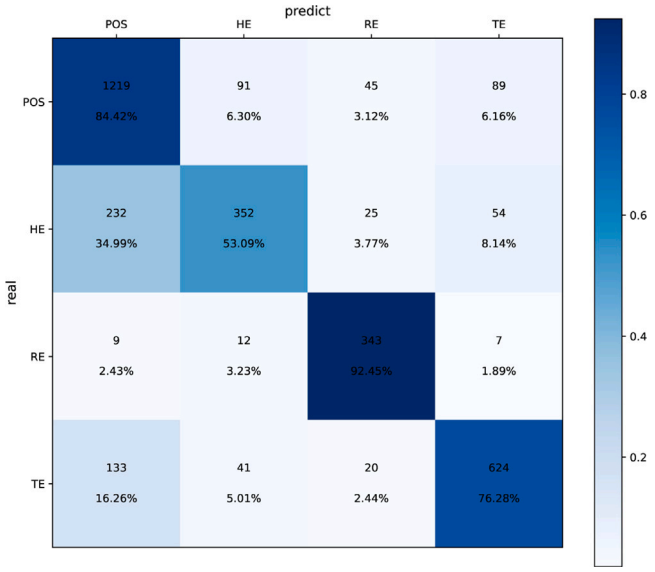


Fig. 9. The confusion matrix.

Servo motor assembly), (First time airbag light, attr_failure, Alarm), and (Button "AUTO", part_of, Stuck), which are logically consistent and backed by higher confidence scores. Nevertheless, it demonstrates less robust predictive ability for triples characterized by ambiguous definitions, e.g., (Iron core grounding wire, part_of, Bolt), (Left A-pillar, part_failure, Smoke), and (Throttle valve, part_failure, Striped deformation). Notably, there are isolated occurrences where triples that adhere to knowledge graph conventions are incorrectly classified as errors, exemplified by (Right-front wheel, part_failure, Noise). This shortcoming stems from the challenge in semantic interpretation when faced with triples entailing multiple complex relationships, thus reducing predictive efficacy. This case study underscores that while our model generally demonstrates robust capacity in discerning accurate error types in most situations, it occasionally struggles to differentiate between closely-related yet discrete errors, particularly those involving head and tail entities. These findings point to potential areas for enhancement, specifically in managing the complexities inherent in triple representations within industrial knowledge graphs, where relationships may be more nuanced and contextually dependent.

7. Conclusion

In this paper, we proposes a method for KG triplets validation, which combines a triple-based strategy that contains the TAAPPNP, RGCN, and PGGN. It achieves good results on both open-domain datasets and industrial KG datasets. Through comparative experiments in various aspects such as model comparison, parameter analysis, and dataset comparison. Compared to the baselines, our TTMNM in this article has improved by about 5%–15% in F1 scores. It can be concluded that the proposed model in this paper has good applicability and suitability. In particular, TAAPPNP and PGGN plays a crucial role in graph reasoning algorithms. In future work, we will apply this algorithm to various downstream tasks of different KGs for inference and application. The distribution of the dataset also has very different predictive effects on the model, and future work will also include improvements to the negative example data generation algorithm.

CRediT authorship contribution statement

Geng Zhang: Conceptualization, Data curation, Formal analysis, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Yu-Jie Xiong:** Conceptualization, Methodology,

Validation, Writing – review & editing. **Jian-Peng Hu:** Conceptualization, Methodology, Writing – review & editing. **Chun-Ming Xia:** Conceptualization, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was funded by National Natural Science Foundation of China, 62006150.

Data availability

Data will be made available on request.

References

Abu-Salih, B., 2021. Domain-specific knowledge graphs: A survey. J. Netw. Comput. Appl. 185, 103076.

Ahmad, W.U., Peng, N., Chang, K.-W., 2021. GATE: graph attention transformer encoder for cross-lingual relation and event extraction. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 14, pp. 12462–12470.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O., 2013. Translating embeddings for modeling multi-relational data. Adv. Neural Inf. Process. Syst. 26.

Brin, S., 1998. The PageRank citation ranking: bringing order to the web. In: Proceedings of ASIS, 1998, vol. 98, pp. 161–172.

Cao, Z., Xu, Q., Yang, Z., Cao, X., Huang, Q., 2021. Dual quaternion knowledge graph embeddings. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 8, pp. 6894–6902.

Chao, L., He, J., Wang, T., Chu, W., 2021. PairRE: knowledge graph embeddings via paired relation vectors. In: Zong, C., Xia, F., Li, W., Navigli, R. (Eds.), Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics, pp. 4360–4369.

Chen, Z.-M., Yeh, M.-Y., Kuo, T.-W., 2021. PASSLEAF: a pool-based semi-supervised learning framework for uncertain knowledge graph embedding. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 5, pp. 4019–4026.

Ebisu, T., Ichise, R., 2018. TorusE: Knowledge graph embedding on a Lie group. In: McIlraith, S.A., Weinberger, K.Q. (Eds.), Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018. AAAI Press, pp. 1819–1826.

Heindorf, S., Potthast, M., Stein, B., Engels, G., 2016. Vandalism detection in wikidata. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. pp. 327–336.

Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G.d., Gutierrez, C., Kirrane, S., Gayo, J.E.L., Navigli, R., Neumaier, S., et al., 2021. Knowledge graphs. ACM Comput. Surv. 54 (4), 1–37.

Ji, G., He, S., Xu, L., Liu, K., Zhao, J., 2015. Knowledge graph embedding via dynamic mapping matrix. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 687–696.

Ji, S., Pan, S., Cambria, E., Marttinen, P., Philip, S.Y., 2021. A survey on knowledge graphs: Representation, acquisition, and applications. IEEE Trans. Neural Netw. Learn. Syst. 33 (2), 494–514.

Jia, S., Xiang, Y., Chen, X., Wang, K., 2019. Triple trustworthiness measurement for knowledge graph. In: The World Wide Web Conference. pp. 2865–2871.

Klicpera, J., Bojchevski, A., Günnemann, S., 2019. Predict then propagate: graph neural networks meet personalized pagerank. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net.

Lao, N., Cohen, W.W., 2010. Fast query execution for retrieval models based on path-constrained random walks. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '10, Association for Computing Machinery, New York, NY, USA, pp. 881–888.

Li, Y., Li, X., Lei, M., 2020. CTransE: an effective information credibility evaluation method based on classified translating embedding in knowledge graphs. In: Hartmann, S., Küng, J., Kotsis, G., Tjoa, A.M., Khalil, I. (Eds.), Database and Expert Systems Applications. Springer International Publishing, Cham, pp. 287–300.

Lin, Y., Liu, Z., Luan, H., Sun, M., Rao, S., Liu, S., 2015a. Modeling relation paths for representation learning of knowledge bases. In: Conference on Empirical Methods in Natural Language Processing. pp. 327–336.

Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X., 2015b. Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 29, no. 1.

Moon, C., Harenberg, S., Slankas, J., Samatova, N., 2017. Learning contextual embeddings for knowledge graph completion. PACIS, In: Pacific Asia Conference on Information Systems, vol. 10, no. 3132847.3133095.

Nathani, D., Chauhan, J., Sharma, C., Kaul, M., 2019. Learning attention-based embeddings for relation prediction in knowledge graphs. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics.

Nguyen, H.L., Vu, D.T., Jung, J.J., 2020. Knowledge graph fusion for smart systems: A survey. Inf. Fusion 61, 56–70.

Schlichtkrull, M., Kipf, T.N., Bloem, P., Van Den Berg, R., Titov, I., Welling, M., 2018. Modeling relational data with graph convolutional networks. In: The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15. Springer, pp. 593–607.

Shang, C., Tang, Y., Huang, J., Bi, J., He, X., Zhou, B., 2019. End-to-end structure-aware convolutional networks for knowledge base completion. In: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence. AAAI'19/IAAI'19/EAAI'19, AAAI Press.

Shi, Y., Lu, Y., Yang, R., Yang, Y., 2022. Knowledge graph completion technology research. In: 2022 International Conference on Computing, Robotics and System Sciences. ICRSS, IEEE, pp. 68–73.

Sidi, F., Panahy, P.H.S., Affendey, L.S., Jabar, M.A., Ibrahim, H., Mustapha, A., 2012. Data quality: A survey of data quality dimensions. In: 2012 International Conference on Information Retrieval & Knowledge Management. IEEE, pp. 300–304.

Suchanek, F.M., Kasneci, G., Weikum, G., 2007. Yago: a core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web. pp. 697–706.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., et al., 2017. Graph attention networks. Stat 1050 (20), 10–48550.

Wang, B., Meng, Q., Wang, Z., Wu, D., Che, W., Wang, S., Chen, Z., Liu, C., 2022. InterHT: knowledge graph embeddings by interaction between head and tail entities. CoRR, abs/2202.04897, arXiv:2202.04897.

Wang, Z., Zhang, J., Feng, J., Chen, Z., 2014. Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28, no. 1.

Wei, Z., Su, J., Wang, Y., Tian, Y., Chang, Y., 2020. A novel cascade binary tagging framework for relational triple extraction. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, pp. 1476–1488.

Weikum, G., Dong, X.L., Razniewski, S., Suchanek, F., et al., 2021. Machine knowledge: Creation and curation of comprehensive knowledge bases. Found. Trends Databases 10 (2–4), 108–490.

Xiao, H., Gao, J., Li, Q., Ma, F., Su, L., Feng, Y., Zhang, A., 2016. Towards confidence in the truth: a bootstrapping based truth discovery approach. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '16, Association for Computing Machinery, New York, NY, USA, pp. 1935–1944.

Xiao, H., Huang, M., Hao, Y., Zhu, X., 2015. TransA: An adaptive approach for knowledge graph embedding. arXiv preprint arXiv:1509.05490.

Xie, R., Liu, Z., Lin, F., Lin, L., 2018. Does william shakespeare really write hamlet? Knowledge representation learning with confidence. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1.

Xie, Z., Zhou, G., Liu, J., Huang, J.X., 2020. ReInceptionE: relation-aware inception network with joint local-global structural information for knowledge graph embedding. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, pp. 5929–5939.

Yang, P., Cong, X., Sun, Z., Liu, X., 2021. Enhanced language representation with label knowledge for span extraction. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, pp. 4623–4635.

Yu, L., Luo, Z., Liu, H., Lin, D., Li, H., Deng, Y., 2022. TripleRE: knowledge graph embeddings via tripled relation vectors. CoRR, abs/2209.08271, arXiv:2209.08271.

Zhang, X., Yang, Q., Xu, D., 2022. TranS: transition-based knowledge graph embedding with synthetic relation representation. In: Goldberg, Y., Kozareva, Z., Zhang, Y. (Eds.), Findings of the Association for Computational Linguistics. EMNLP 2022, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, pp. 1202–1208.

Zhao, Y., Liu, J., 2019. Scef: A support-confidence-aware embedding framework for knowledge graph refinement. arXiv preprint arXiv:1902.06377.